Content Protection for Recordable Media Specification

"Content-Copy"

Intel Corporation
International Business Machines Corporation
Matsushita Electric Industrial Co., Ltd.
Toshiba Corporation

Revision 0.9
October 14, 1999

This page is intentionally left blank.

Preface

Notice

THIS DOCUMENT IS PROVIDED "AS IS" WITH NO WARRANTIES WHATSOEVER, INCLUDING ANY WARRANTY OF MERCHANTABILITY, NONINFRINGEMENT, FITNESS FOR ANY PARTICULAR PURPOSE, OR ANY WARRANTY OTHERWISE ARISING OUT OF ANY PROPOSAL, SPECIFICATION OR SAMPLE. IBM, Intel, MEI, and Toshiba disclaim all liability, including liability for infringement of any proprietary rights, relating to use of information in this specification. No license, express or implied, by estoppel or otherwise, to any intellectual property rights are granted herein. This document is an intermediate draft for comment only and is subject to change without notice. Readers should not design products based on this document.

Intellectual Property

Implementation of this specification requires a license from the 4C Entity, LLC.

Contact Information

Please address inquiries, feedback, and licensing requests to License Management Incorporated (LMI):

- Licensing inquiries and requests should be addressed to cprm-licensing@lmicp.com.
- Feedback on this specification should be addressed to cprm-comment@lmicp.com.

The URL for the 4C Entity, LLC web site is http://www.4Centity.com.

This page is intentionally left blank.

Table of Contents

List of Figures

This page is intentionally left blank.

List of Tables

This page is intentionally left blank.

Chapter 1
Introduction

1.

1.1 Purpose and Scope

The *Content Protection for Recordable Media Specification* defines a renewable method for protecting entertainment content when recorded on physical media. The types of physical media supported specifically include, but are not limited to, recordable DVD media and Flash memory. Content Protection for Recordable Media (CPRM) is an integral part of an overall system for protecting entertainment content against unauthorized copying, known as the Content Protection System Architecture (see the corresponding reference in Section 1.4).

The use of this specification and access to the intellectual property and cryptographic materials required to implement it will be the subject of a license. A license authority referred to as the 4C Entity, LLC is responsible for establishing and administering the content protection system based in part on this specification.

1.2 Overview

The CPRM technology is designed to meet the following criteria:

- It meets the content owners' requirements for robustness and system renewability.
- All physical copies of the content are encrypted.
- It is applicable for both audio and video entertainment content.
- It is equally suitable for implementation on PCs and CE devices.
- It is applicable to different media types.

The system is based on the following technical elements:

- Key management for interchangeable media
- Content encryption
- Media based renewability

Figure 1-1 shows an illustrative example of how the system operates. The actual details of component storage and key management will vary with different types of DVD and other supported media, as well as with different applications, as described in subsequent chapters.

Step 1a. The 4C Entity, LLC provides an individual set of secret device keys to the device manufacturer for inclusion into each device produced.

Step 1b. Media manufacturers place an identifier and Media Key Block generated by the 4C Entity, LLC on each piece of compliant media.

Step 2. When compliant media is placed within a compliant drive or player/recorder, a secret Media Key is generated by the device using its secret keys and the Media Key Block stored on the media itself. The same secret Media Key is generated regardless of which compliant device is used to access the media.

Step 3. Content stored on the media is encrypted/decrypted by a Content Key derived from a one-way function of a secret Title Key and the copy control information (CCI) associated with the content. The Title Key is

encrypted and stored on the media using a key derived from a one-way function of the Media Key and Media ID. Again, actual details of key management can vary among different applications, as described in subsequent chapters.

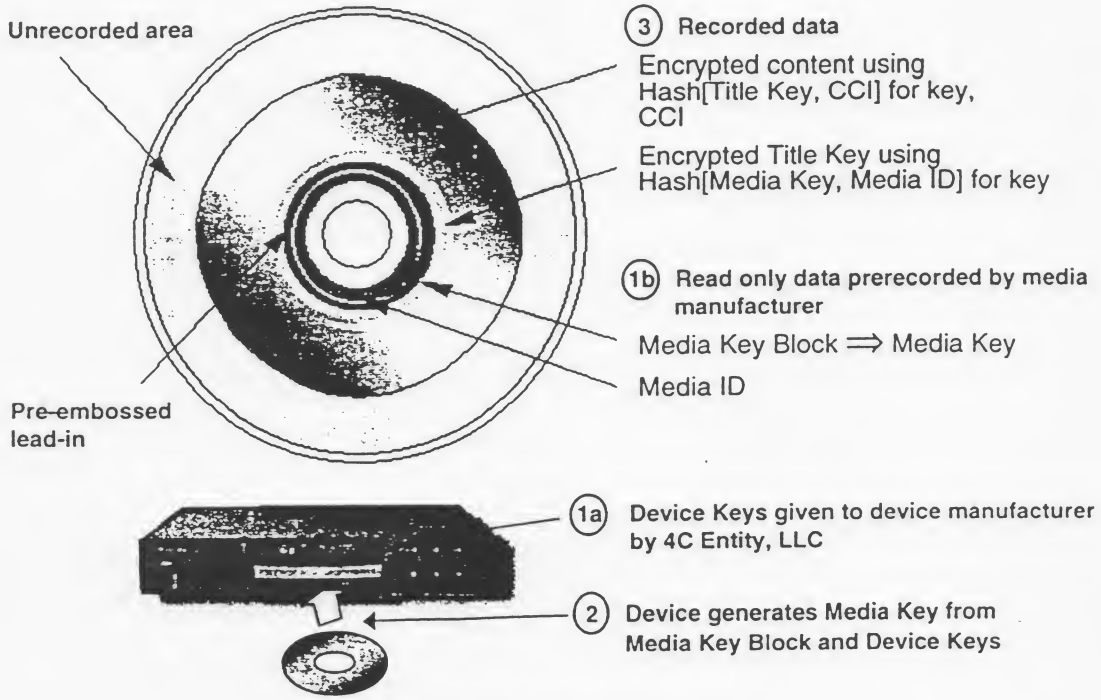

Figure 1-1 – CPRM Illustrative Example

1.3 Organization of this Document

This specification is organized as follows:

- Chapter 1 provides an overview of CPRM.
- Chapter 2 lists abbreviations and acronyms used in this document.
- Chapter 3 describes the media independent components of CPRM.
- Chapter 4 describes the application of CPRM to recordable DVD media.

1.4 References

This specification shall be used in conjunction with the following publications. When the publications are superceded by an approved revision, the revision shall apply.

4C Entity, LLC. *[CPRM license agreement, available soon]*

4C Entity, LLC. *[Content Protection System Architecture White Paper, available soon]*

4C Entity, LLC. *DVD-Audio Content Scramble System, Authenticator on DVD-ROM Drive, Version 1.0*

4C Entity, LLC. *DVD-Audio Content Scramble System, Authenticator on Decoder Card, Version 1.0*

DVD Forum. *DVD Specifications for Rewritable Disc Version 1.0* (Part 1 and 2)

DVD Forum, *DVD Specifications for Rewritable Disc Version 2.0* (Part 1 and 2)

DVD Forum, *DVD Specifications for Rewritable Disc and Re-recordable Disc (Part 3: Video Recording), Version 1.0*

Mt. Fuji Commands for Multimedia Devices

National Institute of Standards and Technology (NIST), *Security Requirements for Cryptographic Modules*, FIPS Publication 140-1, April 14, 1982

Secure Digital Music Initiative (SDMI), *SDMI Portable Device Specification Version 1.0*

1.5 Future Directions

Due to its robust cryptography, key management, and renewability mechanisms, CPRM is expected to address the needs for protected use of audio and video content on recordable media well into the future. Over the course of time, it is expected that CPRM will develop and expand, through updated revisions of this specification, to address additional media types, application formats, and usage models. Some extensions in progress at the time of this release are:

- Application of CPRM to additional media types:
  - Compact Flash media
  - SD Memory Card media.
  - Hard disk drives
- Mechanisms to facilitate SDMI-style "check-in/check-out" and "move" operations on DVD media.

1.6 Notation

1.6.1 Numerical Values

This specification uses three different representations for numerical values. Decimal numbers are represented without any special notation. Binary numbers are represented as a string of binary (0, 1) digits followed by a subscript 2 (e.g., 1010_2). Hexadecimal numbers are represented as a string of hexadecimal (0..9, A..F) digits followed by a subscript 16 (e.g., $3C2_{16}$).

1.6.2 Bit and Byte Ordering

Certain data values or parts of data values are interpreted as an array of bits. Unless explicitly noted otherwise, bit positions within an n-bit data value are numbered such that the least significant bit is numbered 0 and the most significant bit is numbered n-1.

Unless explicitly noted otherwise, big-endian ordering is used for multiple-byte values, meaning that byte 0 is the most significant byte.

1.6.3 Operations

The following notation will be used for bitwise and arithmetic operations:

$[x]_{msb\_z}$	The most significant z bits of x.
$[x]_{lsb\_z}$	The least significant z bits of x.
$[x]_{y:z}$	The inclusive range of bits between bit y and bit z in x.
$x \| y$	Ordered concatenation of x and y.
$x \oplus y$	Bit-wise Exclusive-OR (XOR) of two strings x and y.

x + y	Modular addition of two strings x and y.
x × y	Multiplication of x and y.
x – y	Subtraction of y from x.

The following assignment and relational operators will be used:

=	Assignment
==	Equal to
!=	Not equal to

Chapter 2
Abbreviations and Acronyms

2. Alphabetical List of Abbreviations and Acronyms

The following abbreviations and acronyms are used in this document:

4C	4 Companies (IBM, Intel, MEI, and Toshiba)
AV	Audio-Visual
Addr.	Address
AGID	Authentication Grant ID
AKE	Authentication and Key Exchange
APS	Analog Protection System
APSTB	Analog Protection System Trigger Bits
ASCII	American Standard Code for Information Interchange
ATA	AT Attachment
ATAPI	ATA Packet Interface
Auth.	Authentication
BCA	Burst Cutting Area
C-CBC	Converted Cipher Block Chaining
C2	Cryptomeria Cipher
CCI	Copy Control Information
CD	Compact Disc
CE	Consumer Electronics
Cmd.	Command
CPRM	Content Protection for Recordable Media
CGMS	Copy Generation Management System
CSS2	DVD-Audio Content Scramble System
DVD	Digital Versatile Disc
DVD-RAM	Digital Versatile Disc – Rewritable
DVD-R	Digital Versatile Disc – Recordable
DVD-RW	Digital Versatile Disc – Re-recordable
ECB	Electronic Codebook
ECC	Error Correction Code
FAT	File Allocation Table
FM	Flash Media

FIPS	Federal Information Processing Standards
ID	Identifier
LCM	Licensed Compliant Module
LLC	Limited Liability Company
lsb	Least Significant Bit
MAC	Message Authentication Code
MKB	Media Key Block
MPEG	Moving Picture Expert Group
msb	Most Significant Bit
PC	Personal Computer
PCMCIA	Personal Computer Memory Card International Association
PD	Portable Device
PES	Packetized Elementary Stream
PM	Portable Media
RDI	Real-time Data Information
RTR	Real Time Recording
SD	Secure Digital
SDMI	Secure Digital Music Initiative
XOR	Exclusive-OR

Chapter 3
CPRM Common Elements

3. Introduction

CPRM provides a comprehensive set of encryption, key management, and renewability mechanisms for protecting content stored on recordable media. This chapter describes common mechanisms that are used by CPRM for all media types and applications. The mechanisms are described here in isolation; their specific uses, along with other details that pertain to specific media types and applications, are described in later chapters.

3.1 Cryptographic Algorithms

All cryptographic algorithms used for CPRM are based on the C2 block cipher. This section describes the C2 block cipher algorithm, as well as other common CPRM cryptographic algorithms built on top of that cipher.

3.1.1 C2 Block Cipher in Electronic Codebook (ECB) Mode

[Note: A detailed description of the C2 block cipher algorithm in ECB mode is under review and will be available soon]

In this document, encryption with the C2 cipher in Electronic Codebook (ECB) mode is represented by the function

C2_E(k, d)

where k is a 56-bit key, d is 64-bit data value to be encrypted, and C2_E returns the 64-bit result.

Decryption using the C2 cipher in ECB mode is represented by the function

C2_D(k, d)

where k is a 56-bit key, d is a 64-bit data value to be decrypted, and C2_D returns the 64-bit result.

3.1.2 C2 Block Cipher in Converted Cipher Block Chaining (C-CBC) Mode

[Note: A detailed description of the C2 block cipher algorithm in C-CBC mode is under review and will be available soon]

To increase the robustness of the baseline C2 cipher, Converted Cipher Block Chaining (C-CBC) is used for the encryption and decryption of content protected by CPRM.

In this document, encryption with the C2 cipher in C-CBC mode is represented by the function

C2_ECBC(k, d)

where k is a 56-bit key, d is a frame of data to be encrypted, and C2_ECBC returns the encrypted frame.

Decryption using the C2 cipher in C-CBC mode is represented by the function

C2_DCBC(k, d)

where k is a 56-bit key, d is a frame of data to be decrypted, and C2_DCBC returns the decrypted frame.

The size of the frame of data to be encrypted or decrypted depends on the particular video or audio application format, and is defined for each in the corresponding section of this document.

3.1.3 C2 Hash Function

CPRM uses a hashing procedure based on the C2 encryption algorithm. This procedure is called the C2 Hash Function, and is represented by the function

C2_H(d)

where d is input data with a length that is a multiple of 8 bytes, and C2_H returns the 64-bit result.

Figure 3-1 depicts the hashing procedure.

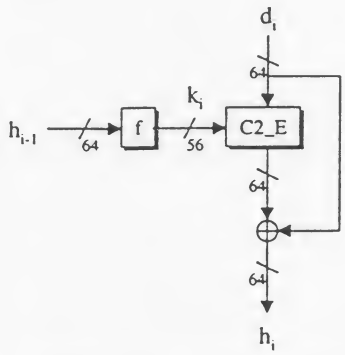

Figure 3-1 – C2 Hash Function

The input data d is divided into n 64-bit blocks, represented as $d_1, d_2, \ldots d_n$.

A conversion function f is defined as

$f(x) = [x]_{lsb\_56}$

where x is a 64-bit input data value d.

A 64-bit fixed initial value h_0 is provided to licensees by the 4C Entity, LLC.

The following are calculated iteratively for i from 1 to n:

$k_i = f(h_{i-1})$

and

$h_i = C2\_E(k_i, d_i) \oplus d_i$.

The value h_n is the final result of the hash, i.e. $C2\_H(d) = h_n$.

3.1.4 C2 One-way Function

CPRM uses a one-way function based on the C2 encryption algorithm. This function is called the C2 One-way Function, and is represented by

$C2\_G(d_1, d_2)$

where d_1 is a 56-bit input data value, d_2 is a 64-bit input data value, and C2_G returns the 64-bit result.

Figure 3-2 depicts the one-way function.

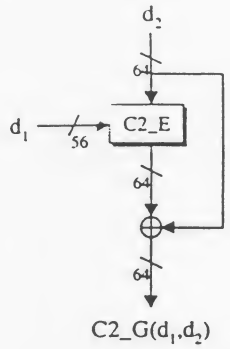

Figure 3-2 – C2 One-way Function

The one-way function result is calculated as

$$C2\_G(d_1, d_2) = C2\_E(d_1, d_2) \oplus d_2.$$

3.1.5 C2 Random Number Generator

The C2 Random Number Generator is a random number generator based on the C2 One-way Function. This random number generator (or one of equivalent or higher quality[1]) must be used for CPRM. Figure 3-3 shows the seed generation and random number generation processes for the C2 Random Number Generator.

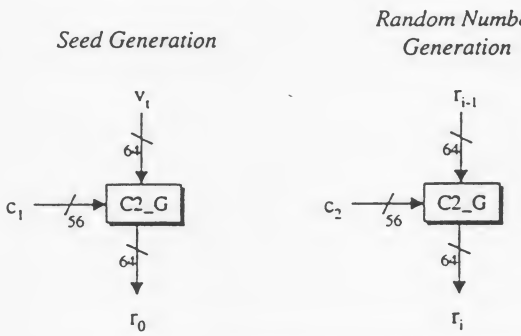

Figure 3-3 – C2 Random Number Generator

The seed generation process must occur before a device first generates a random number after being powered on. A 64-bit value v_t is acquired from a time-varying source. The source can either be a non-volatile memory register that is updated each time it is sampled, or a free-run counter. If a non-volatile register is used, the manufacturer shall assign an initial value to the register that is statistically unique per device. If a free-run counter is used, it shall always be initialized at power-up to a manufacturer-assigned (fixed) value that is statistically unique per device. The confidentiality of v_t must be maintained, as indicated in the Robustness Rules section of the CPRM license document.

56-bit constant values c_1 and c_2 are assigned to licensees by the 4C Entity, LLC.

The 64-bit seed r_0 is calculated as

$$r_0 = C2\_G(c_1, v_t).$$

---

[1] Other FIPS-140 compliant (using the tests described in FIPS-140 section 4.11.1) random number generators supporting run-time non-correlated input as well as a seed value generated at either runtime or manufacture time by a physical random process may be used.

After seed generation, the C2 Random Number Generator can output 64-bit random numbers r_i ($i=1,2,\ldots$), which are each generated as

$$r_i = C2\_G(c_2, r_{i-1}).$$

3.2 Common Cryptographic Key Management

Figure 3-4 depicts a common cryptographic key management procedure, which is a part of CPRM protection for all media types and applications.

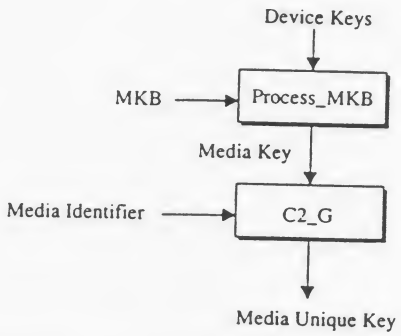

Figure 3-4 – Common CPRM Cryptographic Key Management Procedure

- Device Keys ($K_{d\_0}, K_{d\_1}, \ldots, K_{d\_n-1}$) are used to decrypt one or more elements of a Media Key Block (MKB), in order to extract a secret Media Key (K_m).
- K_m and a Media Identifier (ID_{media}) are combined, using the C2 One-way Function, to produce a Media Unique Key (K_{mu}).

Table 3-1 lists the elements involved in this process, along with their sizes.

Table 3-1 – Common Cryptographic Key Management Elements

Key or Variable	Size
Device Keys ($K_{d\_0}, K_{d\_1}, \ldots, K_{d\_n-1}$)	56 bits each
Media Key Block (MKB)	Variable, multiple of 8 bytes
Media Key (K_m)	56 bits
Media Identifier (ID_{media})	64 bits
Media Unique Key (K_{mu})	56 bits

The remainder of this section describes this common key management procedure in detail. Note that the procedure is described here in isolation; its use as part of CPRM for different media types and applications is described in later chapters.

3.2.1 Calculation of the Media Key (K_m)

3.2.1.1 Device Keys

Each CPRM compliant device is given an individual set of secret Device Keys when manufactured. These keys are provided by the 4C Entity, LLC, and are for use in processing the MKB to calculate K_m.

Each device receives n Device Keys, which are referred to as K_{d_i} (i=0,1,…,n-1). For each Device Key there is an associated Column and Row value, referred to as C_{d_i} and R_{d_i} (i=0,1,…,n-1) respectively. Column and Row values start at 0. For a given device, no two Device Keys will have the same associated Column value (in other words, a device will have at most one Device Key per Column). It is possible for a device to have some Device Keys with the same associated Row values. The number of Device Keys that are given to each device and the range of Column and Rows values that are possible are defined separately for each device type in the corresponding chapter of this document.

The confidentiality of the Device Keys and their associated Column and Row values must be maintained, as indicated in the Robustness Rules section of the CPRM license document.

3.2.1.2 Media Key Block (MKB)

CPRM's cryptographic key management scheme uses the Media Key Block (MKB) to enable system renewability. The MKB is generated by the 4C Entity, LLC, and allows all compliant devices, each using their unique set of secret Device Keys, to calculate the same K_m. If a set of Device Keys is compromised in a way that threatens the integrity of the system, new media can be released containing an updated MKB. The updated MKB will cause a device with the compromised set of Device Keys to calculate a different K_m than is computed by the remaining compliant devices.

An MKB is formatted as a sequence of contiguous Records. Each Record begins with a one-byte Record Type field, followed by a three-byte Record Length field. The Record Type field value indicates the type of the Record, and the Record Length field value indicates the number of bytes in the Record, including the Record Type and the Record Length fields themselves. Record lengths are always multiples of 4 bytes. The Record Type and Record Length fields are never encrypted. Subsequent fields in a Record may be encrypted (by the C2 cipher in ECB mode), depending on the Record Type.

Using its Device Keys, a device calculates K_m by processing Records of the MKB one-by-one, in order, from first to last. Except where explicitly noted otherwise, a device must process every Record of the MKB. The device must not make any assumptions about the length of Records, and must instead use the Record Length field value to go from one Record to the next. If a device encounters a Record with a Record Type field value it does not recognize, it ignores that Record and skips to the next. For some Records, processing will result in the calculation of a K_m value. Processing of subsequent Records may update the K_m value that was calculated previously. After processing of the MKB is completed, the device uses the most recently calculated K_m value as the final value for K_m (i.e. the output of Process_MKB in Figure 3-4).

If a device correctly processes an MKB using device keys that are revoked by that MKB, the resulting final K_m shall have the special value 00000000000000_{16}. This special value shall never be an MKB's correct final K_m value, and can therefore always be taken as an indication that the device's keys are revoked. Device behavior in this situation is implementation defined. As an example, a device could exhibit a special diagnostic code, as information to a service technician.

The following subsections describe the currently defined Record types, and how a device processes each.

3.2.1.2.1 Verify Media Key Record

Table 3-2 shows the format of a *Verify Media Key* Record.

Table 3-2 – *Verify Media Key* Record Format

Byte \ Bit	7	6	5	4	3	2	1	0
0	Record Type: 81_{16}							
1	Record Length: $00000C_{16}$							
2								
3								
4	Verification Data (D_v): C2_E(K_m, $DEADBEEF_{16}$ ‖ $XXXXXXXX_{16}$)							
...								
11								

A properly formatted MKB shall have exactly one *Verify Media Key* Record as its first Record. The final 8 bytes of the Record contain the value

$$C2\_E(K_m, DEADBEEF_{16} \| XXXXXXXX_{16})$$

where K_m is the correct final Media Key value, and $XXXXXXXX_{16}$ is an arbitrary 4-byte value.

The presence of the *Verify Media Key* Record in an MKB is mandatory, but he use of the Record by a device is optional.

As an optimization, a device may attempt to decrypt D_v using its current K_m value during the processing of subsequent records, checking each time for the condition

$$[C2\_D(K_m, D_v)]_{msb\_32} == DEADBEEF_{16}$$

where K_m is the current Media Key value.

If this condition is true, the device has already calculated the correct final K_m value, and can therefore stop processing the MKB.

Also (or alternatively), a device could check the same condition after processing the entire MKB, in order to determine if it has calculated the correct final K_m. Failure to calculate the correct K_m after processing the entire MKB could be the result of data or calculation errors, or of the device's keys having been revoked (or both). Note that these two cases can generally be distinguished, since a device with revoked keys that correctly processes the MKB will calculate an incorrect final K_m with the special value 00000000000000_{16}.

3.2.1.2.1 Verify Media Key Record

Table 3-2 shows the format of a *Verify Media Key* Record.

Table 3-2 – *Verify Media Key* Record Format

Byte \ Bit	7	6	5	4	3	2	1	0
0	Record Type: 81_{16}							
1	Record Length: $00000C_{16}$							
2								
3								
4	Verification Data (D_v): C2_E(K_m, $DEADBEEF_{16}$ ‖ $XXXXXXXX_{16}$)							
...								
11								

A properly formatted MKB shall have exactly one *Verify Media Key* Record as its first Record. The final 8 bytes of the Record contain the value

$C2\_E(K_m, DEADBEEF_{16} \| XXXXXXXX_{16})$

where K_m is the correct final Media Key value, and $XXXXXXXX_{16}$ is an arbitrary 4-byte value.

The presence of the *Verify Media Key* Record in an MKB is mandatory, but he use of the Record by a device is optional.

As an optimization, a device may attempt to decrypt D_v using its current K_m value during the processing of subsequent records, checking each time for the condition

$[C2\_D(K_m, D_v)]_{msb\_32} == DEADBEEF_{16}$

where K_m is the current Media Key value.

If this condition is true, the device has already calculated the correct final K_m value, and can therefore stop processing the MKB.

Also (or alternatively), a device could check the same condition after processing the entire MKB, in order to determine if it has calculated the correct final K_m. Failure to calculate the correct K_m after processing the entire MKB could be the result of data or calculation errors, or of the device's keys having been revoked (or both). Note that these two cases can generally be distinguished, since a device with revoked keys that correctly processes the MKB will calculate an incorrect final K_m with the special value 00000000000000_{16}.

3.2.1.2.2 Calculate Media Key Record

Table 3-3 shows the format of a *Calculate Media Key* Record.

Table 3-3 – *Calculate Media Key* Record Format

	Bit / Byte	7	6	5	4	3	2	1	0
	0	Record Type: 01_{16}							
	1	Record Length							
	2								
	3								
	4	Reserved							
	...								
	7								
	8	Column							
	9	Generation: 000001_{16}							
	10								
	11								
Encrypted Key Data	12	Encrypted Key Data for Row 0 (D_{kc_0})							
	...								
	19								
	20	Encrypted Key Data for Row 1 (D_{kc_1})							
	...								
	27								
	28	.							
	...	.							
	Length-1	.							

A properly formatted MKB shall have exactly one *Calculate Media Key* Record. Devices must ignore any *Calculate Media Key* Records encountered after the first one in an MKB. The use of the Reserved field is currently undefined, and it is ignored. The Generation field shall contain 000001_{16} for the first generation. The Column field indicates the associated Column value for the Device Key to be used with this Record, as indicated below. Bytes 12 through the last byte of the Record contain Encrypted Key Data. The first eight bytes of the Encrypted Key Data correspond to Device Key Row 0, the next eight bytes correspond to Device Key Row 1, and so forth.

Before processing the Record, the device checks that both of the following conditions are true:

Generation == 000001_{16}

and

the device has a Device Key with associated Column value C_{d_i} == Column, for some i.

If either of these conditions is false, the device ignores the rest of the Record.

Otherwise, using the value i from the condition above, and r = R_{d_i}, the device calculates:

$K_m = [C2\_D(K_{d\_i}, D_{ke\_r})]_{lsb\_56}$

where K_{d_i} is the i^{th} Device Key's value, D_{ke_r} is the 64-bit value starting at byte offset $r \times 8$ within the Record's Encrypted Key Data, and K_m becomes the current Media Key value.

It is not necessary for a first generation device to verify that Record Length is sufficient to index into the Encrypted Key Data. First generation devices are assured that the Encrypted Key Data contains a value corresponding to their Device Key's associated Row value.

3.2.1.2.3 Conditionally Calculate Media Key Record

Table 3-4 shows the format of a *Conditionally Calculate Media Key* Record.

Table 3-4 – *Conditionally Calculate Media Key* Record Format

	Bit / Byte	7	6	5	4	3	2	1	0
	0	Record Type: 82_{16}							
	1	Record Length							
	2								
	3								
Encrypted Conditional Data (D_{ce})	4	$DEADBEEF_{16}$ (encrypted)							
	...								
	7								
	8	Column (encrypted)							
	9	Generation: 000001_{16} (encrypted)							
	10								
	11								
Doubly Encrypted Key Data	12	Doubly Encrypted Key Data for Row 0 (D_{kde_0})							
	...								
	19								
	20	Doubly Encrypted Key Data for Row 1 (D_{kde_1})							
	...								
	27								
	28								
	...								
	Length-1								

A properly formatted MKB may have zero or more *Conditionally Calculate Media Key* Records. Bytes 4 through 11 of the Record contain Encrypted Conditional Data (D_{ce}). If decrypted successfully, as described below, bytes 4 through 7 contain the value $DEADBEEF_{16}$, byte 8 contains the associated Column value for the Device Key to be used with this Record, and bytes 9 through 11 contain a Generation value of 000001_{16} for the first generation. Bytes 12 through the last byte of the Record contain Doubly Encrypted Key Data. The first eight bytes of the Doubly Encrypted Key Data correspond to Device Key Row 0, the next eight bytes correspond to Device Key Row 1, and so forth.

Using its current K_m value, the device calculates Conditional Data (D_c) as:

$D_c = C2\_D(K_m, D_{cc})$.

Before continuing to process the Record, the device checks that all of the following conditions are true:

$[D_c]_{msb\_32} == DEADBEEF_{16}$

and

$[D_c]_{lsb\_24} == 000001_{16}$

and

the device has a Device Key with associated Column value $C_{d\_i} == [D_c]_{31:24}$ for some i.

If any of these conditions is false, the device ignores the rest of the Record.

Otherwise, using the value i from the condition above, and $r = R_{d\_i}$, the device calculates:

$d = C2\_D(K_m, D_{kde\_r})$

where D_{kde_r} is the 64-bit value starting at byte offset $r \times 8$ within the Record's Doubly Encrypted Key Data,

and then uses the resulting value d to calculate:

$K_m = [C2\_D(K_{d\_i}, d)]_{lsb\_56}$

where K_{d_i} is the i^{th} Device Key's value, and K_m becomes the current Media Key value.

3.2.1.2.4 End of Media Key Block Record

Table 3-5 shows the format of an *End of Media Key Block* Record.

Table 3-5 – *End of Media Key Block* Record Format

Bit / Byte	7	6	5	4	3	2	1	0
0	Record Type: 02_{16}							
1	Record Length: 000004_{16}							
2								
3								

A properly formatted MKB shall contain an *End of Media Key Block* Record. When a device encounters this Record it stops processing the MKB, using whatever K_m value it has calculated up to that point as the final K_m (pending optional checks for correctness of the key, as described previously).

3.2.1.2.5 State Machine for Processing the Media Key Block

[Note: Detailed state machine or pseudo-code to be included soon]

3.2.2 Calculation of the Media Unique Key (K_{mu})

3.2.2.1 Media Identifier (ID_{media})

Each piece of CPRM compliant media (or in some cases, playback device, as specified in this document) shall contain an individual identifier that is readable. This identifier does not need to be secret, but must be stored in a manner that prevents it from being directly altered or replaced, as indicated in the Compliance Rules section of the CPRM license document. If an identifier is 64 bits long, its value can be used directly as ID_{media}. For cases where the identifier is not 64 bits long, a function shall be defined in the corresponding section of this document to convert that identifier to a 64-bit value to be used as ID_{media}.

3.2.2.2 Media Unique Key (K_{mu})

CPRM's cryptographic key management uses a Media Unique Key (K_{mu}) to bind encrypted content to the media (or in some cases, device, as specified in this document) on which it will be played back. K_{mu} is calculated using ID_{media} and the previously calculated K_m, as follows:

$$K_{mu} = [C2\_G(K_m, ID_{media})]_{lsb\_56}$$

3.3 Encryption and Decryption of Content

Content stored in a CPRM recording is encrypted and decrypted using the C2 cipher algorithm in C-CBC mode. The frame size and method for calculating encryption and decryption keys are defined separately for each audio or video application format.

Chapter 4
CPRM for DVD Media Formats

4. Introduction

This chapter specifies details for using CPRM technology to protect entertainment content stored on DVD Rewritable media. The DVD physical format addressed in this chapter is:

- DVD-RAM 4.7 GB:

The DVD application format addressed in this chapter is:

- Video Recording:

These formats are licensable from the DVD Forum, which also publishes specifications describing them in detail (see the corresponding references in Section 1.4). This chapter assumes that the reader is familiar with these formats, as defined in their corresponding specifications.

It is anticipated that CPRM technology will also be applied to other DVD physical and application formats under future extensions to this specification, as authorized by the 4C Entity, LLC. Specific physical formats being considered for possible inclusion in a future revision of this specification include:

- DVD-RAM 2.6GB:
  Currently see DVD Forum, *DVD Specifications for Rewritable Disc Version 1.0* (Part 1 and 2)
- DVD-RW 4.7 GB:
  Currently see DVD Forum, *DVD Specifications for Re-recordable Disc Version 0.9* (Part 1 and 2)
- DVD-R:
  Currently see DVD Forum, *DVD Specifications for Recordable Disc Version 1.0 (Part 1 and 2)*

Specific application formats being considered for possible inclusion in a future revision of this specification include:

- Audio Recording:
  Specification currently a work in progress in DVD Forum
- Digital Stream Recording:
  Specification currently a work in progress in DVD Forum

When CPRM is used for other DVD physical and application formats, the provided protection shall include the common mechanisms described in Chapter 3, and shall provide protection at a level consistent with that provided for the formats currently addressed in this chapter.

4.1 Device Requirements

Each CPRM compliant DVD recording or playing device is given an individual set of 16 secret Device Keys, denoted K_{d_0},K_{d_1},....K_{d_15}. These keys are provided by the 4C Entity, LLC, and are for use in processing the MKB to calculate the Media Key (K_m), as described in Chapter 3. The confidentiality of the Device Keys and their associated Column and Row values must be maintained, as indicated in the Robustness Rules section of the CPRM license document.

4.2 Application of Common CPRM Components to DVD-RAM Media

This section describes location and format details of the common CPRM components described in Chapter 3, when stored on DVD-RAM media. Figure 4-1 gives an overview of the physical locations of these components.

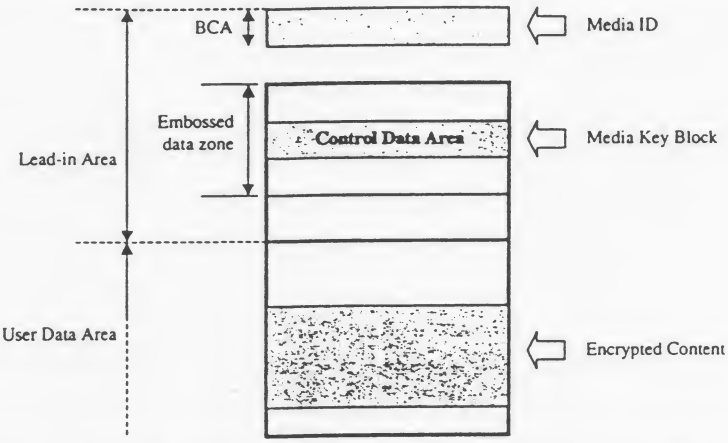

Figure 4-1 – Physical Layout of Common CPRM Components on DVD-RAM Media

- A Media Identifier (ID_{media}) is pre-recorded in the Burst Cutting Area (BCA).
- A Media Key Block (MKB) is pre-recorded in the Embossed data zone of the Lead-in Area.
- Encrypted Content is recorded in the User Data Area.

In addition, other application-specific components related to CPRM may also be stored in the User Data Area, as described later in the section on DVD application formats (Section 4.3).

The remainder of the current section contains further details on the location and format of the Media Identifier and MKB. DVD-RAM media containing a Media Identifier and MKB as described in this section is referred to as CPRM compliant DVD-RAM media.

4.2.1 Media Identifier

[Note: BCA formatting details described in this sub-section are pending DVD-Forum discussion].

CPRM compliant DVD-RAM media shall contain a 64-bit Media Identifier (ID_{media}), which is placed in the Burst Cutting Area (BCA) by the media manufacturer. The BCA can contain multiple blocks of data, called BCA Packs, each containing information for a different use. For CPRM compliant DVD-RAM media, the BCA shall include with a BCA Pack containing a Media Identifier, with format as shown in Table 4-1.

Table 4-1 – Format of Burst Cutting Area Pack Containing the Media Identifier

Byte \ Bit	7	6	5	4	3	2	1	0
0	(msb)							
1	Application ID: 0002_{16}							(lsb)
2	Version: 01_{16}							
3	Data Length: 08_{16}							
4	(msb)							
.	Data: Media Identifier							
11								(lsb)

The Application ID field identifies the use for each BCA Pack, with the value 0002_{16} indicating a Media Identifier. The Version field for the Media Identifier is currently defined as 01_{16}. The Data Length field indicates the length in bytes of the subsequent Data field, which is 08_{16} for the Media Identifier Version 01_{16}. The Media Identifier itself is contained in the Data field, and has the format shown in Table 4-2.

Table 4-2 – Media Identifier Format for DVD-RAM

Byte \ Bit	7	6	5	4	3	2	1	0
0	Manufacturer ID							
1								
2								
3	Serial Number							
4								
5								
6								
7								

The 4C Entity, LLC assigns each licensee a unique 3-byte value to use in the Manufacturer ID field. Each licensee assigns 5-byte values to the Serial Number field that are unique for each piece of compliant DVD-RAM media that it manufactures.

4.2.2 Media Key Block (MKB)

[Note: Control Data Area formatting details described in this sub-section are pending DVD-Forum discussion].

CPRM compliant DVD-RAM media shall contain an MKB and an MKB Descriptor (described later), that are referred to together as the MKB Frame. The media manufacturer places the MKB Frame in the Control Data Area of the lead-in area's embossed data zone. The layout of the Control Data Area is shown in Table 4-3.

Table 4-3 – Layout of Control Data Area

ECC Blocks	Sectors	
	0-1	2-15
0-15	Already Defined	MKB Pack #0
		...
		MKB Pack #15
16-31		MKB Pack #0
		...
		MKB Pack #15
...		...
176-191		MKB Pack #0
		...
		MKB Pack #15

The Control Data Area consists of 192 ECC Blocks of 16 sectors each. The first two sectors (Sectors 0 and 1) of each ECC Block have uses already defined by the DVD Forum. The remaining 14 sectors (Sectors 2 through 15) are available for storage of the MKB Frame. The 192 ECC Blocks of the Control Data Area are logically divided into 12 groups of 16 ECC Blocks each. Each group of 16 ECC Blocks contains identical data, meaning that the data is repeated 12 times for data integrity purposes. Sectors 2 through 15 of each ECC Block form a 28,672-byte data unit referred to as an MKB Pack. In all there are 16 MKB Packs, each repeated 12 times.

The MKB Frame is constructed from the data contained in the first n MKB Packs, where n depends on the size of the MKB Frame, and is at least 1 and at most 16. The bytes in the n MKB Packs are concatenated, in order, to form the MKB Frame. The first n-1 MKB Packs shall be used completely; the n^{th} MKB Pack may end with unused bytes, which are zero-filled. Figure 4-2 shows the formation of an MKB Frame in a case where n is 3.

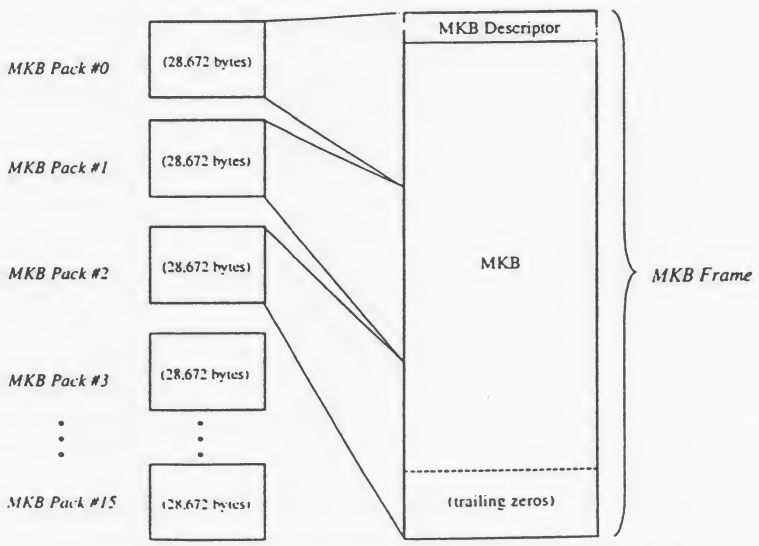

Figure 4-2 – Formation of an MKB Frame from 3 MKB Packs

The MKB Frame begins with a 12-byte MKB Descriptor, which is formatted as shown in Table 4-4.

Table 4-4 – Format of MKB Descriptor

Byte \ Bit	7	6	5	4	3	2	1	0
0	MKB_Hash							
...								
7								
8	Reserved: 0000000000000000_{16}							
...								
15								

The MKB_Hash field contains an 8-byte hash covering the MKB, along with any trailing zeros that may follow it (i.e. covering the entire MKB Frame, except for the MKB Descriptor), and is calculated as

MKB_Hash = C2_H(MKB and trailing zeros).

The MKB_Hash is used to ensure the integrity of the MKB when it is transferred from a drive to a PC host, as described in Section 4.4. The final 8 bytes of the MKB Descriptor are reserved, and shall have the value 0000000000000000_{16}.

The rest of the MKB Frame consists of the MKB itself, which is formatted as described in Section 3.2.1.2, possibly followed by trailing zeros. For the first generation, there may be at most 16 MKB packs, allowing for a maximum MKB size of 16 × 28,672 - 16 = 458,736 bytes. For the first-generation DVD-RAM MKB, 16 Device Key Columns are defined, and a given Column can have at most 4096 Rows.

The number of MKB Packs used to construct the MKB Frame is determined using a field of the Copyright Management Information (CPR_MAI) table. The disc manufacturer pre-records the CPR_MAI table in the each of the sector headers of relative sector numbers 2 through 15 of each ECC Block in the Control Data Area. Table 4-5 shows the format of the CPR_MAI table.

Table 4-5 – CPR_MAI Table Format

Byte \ Bit	7	6	5	4	3	2	1	0
0	Copyright Protection System Type (CPS_TY): 02_{16}							
1	Reserved							
2	CPRM Version: 01_{16}							
3	Total MKB Packs Used							
4	Reserved				CPRM Authentication Control Code			
5	Reserved							

The CPS_TY field contains the value 02_{16}, indicating that the disc contains data structures for CPRM (i.e. is CPRM compliant). Other possible values for CPS_TY are currently reserved. The CPRM Version field value is currently defined as 01_{16}. The Total MKB Packs Used field indicates the number of MKB Packs to be used in constructing the MKB Frame. The CPRM Authentication Control Code field is used in conjunction with the authentication scheme described in Section 4.4.1.

4.3 DVD Application Formats

This section describes the use of CPRM for specific DVD application formats. Details specific to each application format are provided, including the locations of cryptographic elements within the format, and their use in CPRM cryptographic key management and encryption.

4.3.1 Video Recording Format

The Video Recording format is defined by the DVD Forum for real-time recording (on Rewritable and Re-recordable DVD media) of moving pictures and still pictures with associated audio. The Video Recording format is the subject of a license from the DVD Forum, which also publishes a specification describing the format in detail (see the corresponding reference in Section 1.4). This section assumes the reader is familiar with the Video Recording format, as defined in that specification.

4.3.1.1 Stored Data Values Relevant to CPRM

For each disc, the Video Recording format uses a management information file named VR_MANGR.IFO. Included in this file is a 512-byte Video Manger Information Management Table (VMGI_MAT), part of which is used by CPRM to store a 64-bit Encrypted Title Key (K_{te}) and an associated Encrypted Title Key Status (K_{te}_Stat) bit, as shown in Table 4-6.

Table 4-6 – Storage of Encrypted Title Key in VMGI_MAT

Byte \ Bit	7	6	5	4	3	2	1	0
0	(Data defined in Video Recording specification)							
...								
263								
264	Reserved: 0							
265	Reserved: 0							
266	Reserved: 0							
267	Reserved: 0							K_{te}_Stat
268	K_{te}							
...								
275								
276	(Data defined in Video Recording Specification)							
...								
511								

Note that for Video Recording, there is a single K_{te} per disc. The usage of K_{te} is described later. The K_{te}_Stat field indicates the status of the K_{te} field, as shown in Table 4-7.

Table 4-7 – Encoding of K_{te}_Stat Field in VMGI_MAT

K_{te}_Stat value	Status of K_{te} field
0	No valid K_{te} value is present
1	A valid K_{te} value is present

The Video Recording format stores content stream data in stream data files. Content stream data is structured as a sequence of 2048-byte packs, which hold different information depending on the pack type. Real-time Data Information (RDI) packs carry real-time data information. Video packs, Audio packs, and Sub-picture packs carry audio-visual content, and are referred to generically in this document as AV Packs.

RDI packs occur periodically within a content stream (with presentation times at least 0.4 seconds apart), and are used to carry various types of information about the stream. RDI packs are not encrypted. Table 4-8 shows two data fields of an RDI pack that are relevant to CPRM content protection.

Table 4-8 – RDI Pack Data Fields Relevant to CPRM Content Protection

Field name	Description	Size (bits)
CGMS	Copy control information	2
APSTB	Analog protection status	2

The CGMS and APSTB fields are used to indicate the copy control information and analog protection status, respectively, of subsequent AV packs in the recorded content stream. The field values may change in subsequent RDI packs within the stream. Encodings for these two fields are defined in the Video Recording specification.

When the copy control information of incoming content indicates that copying is permitted without restriction, it is recorded without the use of CPRM protection mechanisms. The CGMS field corresponding to that content in the recorded stream is set to indicate that copying is permitted without restriction, and the corresponding AV Packs are not encrypted. Incoming content with copy control information indicating that one generation of copies may be made is recorded with CPRM protection. The CGMS field corresponding to that content in the recorded stream is set to indicate that no more copying is permitted, and *all* of the corresponding AV Packs are encrypted, as described below. Incoming content with copy control information indicating that no copying (or no more copying) is permitted shall not be recorded. Note that the CGMS field has only two defined states for Video Recording recorded streams, one associated with unencrypted content and the other associated with encrypted content. For that reason, no measures are required to protect the recorded CGMS value from malicious tampering.

Table 4-9 shows an encrypted AV Pack.

Table 4-9 – Encrypted AV Pack

	Bit / Byte	7	6	5	4	3	2	1	0
Unencrypted Portion (128 bytes)	0 … 19	(Data defined in Video Recording specification)							
	20			PES_scrambling_ Control					
	21 … 83	(Data defined in Video Recording specification)							
	84 … 91	Title Key Conversion Data (D_{tkc})							
	92 … 127	(Data defined in Video Recording specification)							
Encrypted Portion (1920 bytes)	128 … 2047	Encrypted AV Data (D_{av_e})							

For the Video Recording format, the 2-bit PES_scrambling_control field is set to 11_2 in an encrypted AV Pack, and 00_2 in an unencrypted AV Pack. The use of the 64-bit Title Key Conversion Data (D_{tkc}) is described in the next section. The first 128 bytes of the pack are unencrypted. The final 1920 bytes, referred to as the Encrypted AV Data (D_{av_e}), are encrypted as described in the next section. Before encryption (or after decryption), those same 1920 bytes are referred to as Unencrypted AV Data (D_{av_u}).

Table 4-10 summarizes stored data values that are relevant to CPRM protection of Video Recording formatted content.

Table 4-10 – Video Recording Stored Data Values Relevant to CPRM

Data Value	Size	Storage Location	Comment
Encrypted Title Key (K_{te})	64 bits	VR_MANGR.IFO	One per disc
Encrypted Title Key Status (K_{te}_Stat)	1 bit	VR_MANGR.IFO	Indicates status of K_{te}
APSTB	2 bits	RDI Pack	Analog Protection Status
CGMS	2 bits	RDI Pack	Copy control information
PES_scrambling_control	2 bits	Each encrypted AV Pack	–
Title Key Conversion Data (D_{tkc})	64 bits	Each encrypted AV Pack	–
Encrypted AV Data (D_{av_e})	1920 bytes	Each encrypted AV Pack	C2 C-CBC encryption frame

4.3.1.2 Content Encryption and Decryption

Figure 4-3 illustrates the process for encryption and decryption of Video Recording formatted content on DVD-RAM media.

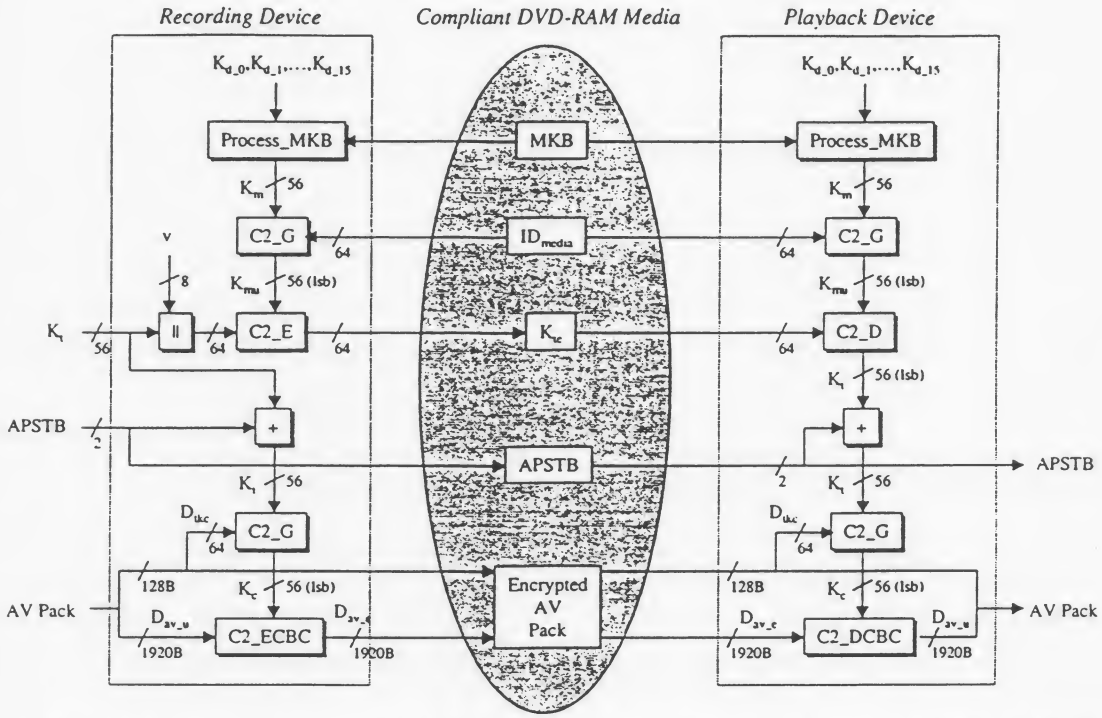

Figure 4-3 – Content Encryption and Decryption for Video Recording Format on DVD-RAM Media

The remainder of this section describes the encryption and decryption processes in detail.

4.3.1.2.1 Content Encryption

The process to encrypt Video Recording formatted content is as follows:

1. Calculate Media Key (K_m):

The Recording Device reads the MKB from the disc, and uses its Device Keys ($K_{d\_0}, K_{d\_1}, \ldots, K_{d\_15}$) to calculate K_m as described in Chapter 3.

2. Calculate Media Unique Key (K_{mu}):

The Recording Device reads the Media Identifier (ID_{media}) from the disc, and calculates K_{mu} as

$$K_{mu} = [C2\_G(K_m, ID_{media})]_{lsb\_56}.$$

3. Generate (or calculate) Title Key (K_t):

For the Video Recording format, a single 56-bit Title Key (K_t) is used for all titles recorded on a given disc.

The Recording Device examines the K_{te}_Stat field of the VMGI_MAT table to determine if an Encrypted Title Key (K_{te}) is already recorded on the disc. If a K_{te} is not already recorded on the disc (K_{te}_Stat == 0), the Recording Device generates K_t using the C2 Random Number Generator, or some other suitable random number generator as defined in Chapter 3.

If a K_{te} is already recorded on the disc (K_{te}_Stat == 1), the Recording Device calculates K_t using the same method that is used by a playback device (see Section 4.3.1.2.2, step 3). Note that the case where the Recording Device uses this step is not illustrated in Figure 4-3.

4. Calculate and record Encrypted Title Key (K_{te}):

If a K_{te} is already recorded on the disc (K_{te}_Stat == 1), this step is skipped.

Otherwise, the Recording Device selects an arbitrary 8-bit value v (any value is acceptable), and calculates K_{te} as

$$K_{te} = C2\_E(K_{mu}, v \| K_t).$$

The Recording Device records K_{te} on the disc (in the VMGI_MAT table), and sets the K_{te}_Stat bit to 1.

5. Calculate intermediate value K_i:

The Recording Device calculates the intermediate 56-bit value K_i by taking

$$K_i = K_t + (0000000000000_{16} \| 00_2 \| APSTB)$$

where + represents addition modulo 2^{56},

and then padding the result to 56 bits by prepending zero-valued bits as needed. The APSTB is recorded on the disc. This step is repeated whenever the APSTB value changes during the recording of encrypted content.

6. Encrypt AV Packs:

For each AV Pack to be encrypted, the Recording Device uses that pack's Title Key Conversion Data (D_{tkc}) to calculate a 56-bit Content Key (K_c) as follows:

$$K_c = [C2\_G(K_i, D_{tkc})]_{lsb\_56}.$$

The resulting K_c value is then used to encrypt that pack's 1920-byte Unencrypted AV Data (D_{av_u}) as follows:

$$D_{av\_e} = C2\_ECBC(K_c, D_{av\_u}).$$

The PES_scrambling_control field of the encrypted AV Pack is set to 11_2.

4.3.1.2.2 Content Decryption

The process to decrypt encrypted Video Recording formatted content is as follows:

1. Calculate Media Key (K_m).

The Playback Device reads the MKB from the disc, and uses its Device Keys (K_{d_0},K_{d_1},…,K_{d_15}) to calculate K_m as described in Chapter 3.

2. Calculate Media Unique Key (K_{mu}):

The Playback Device reads the Media Identifier (ID_{media}) from the disc, and calculates K_{mu} as

$$K_{mu} = [C2\_G(K_m, ID_{media})]_{lsb\_56}.$$

3. Calculate Title Key (K_t):

The Playback Device reads the Encrypted Title Key (K_{te}) from the disc, and calculates K_t as

$$K_t = [C2\_D(K_{mu}, K_{te})]_{lsb\_56}.$$

4. Calculate intermediate value K_i:

The Playback Device calculates the intermediate 56-bit value K_i by taking

$$K_i = K_t + (0000000000000_{16} \| 00_2 \| APSTB)$$

where + represents addition modulo 2^{56}.

and then padding the result to 56 bits by prepending zero-valued bits as needed. This step is repeated whenever the APSTB value is changed in a subsequent RDI Pack.

5. Decrypt AV Packs:

For each AV Pack to be decrypted (i.e. having a PES_scrambling_control field value of 11_2), the Recording Device uses that pack's Title Key Conversion Data (D_{tkc}) to calculate a 56-bit Content Key (K_c) as follows:

$$K_c = [C2\_G(K_i, D_{tkc})]_{lsb\_56}.$$

The resulting K_c value is then used to decrypt that pack's 1920-byte Encrypted Data (D_{av_e}) as follows:

$$D_{av\_u} = C2\_ECBC(K_c, D_{av\_e}).$$

4.4 PC Based System Architecture

CPRM for DVD media formats can be implemented in a PC based system. In such a system, a DVD drive and PC host together act as the Recording Device and/or Playback Device for CPRM protected content. This section describes additional protection mechanisms, as well as host-drive commands, that are used to implement CPRM in a PC based system.

4.4.1 Verifying Integrity of Media Key Block and Media Identifier

When CPRM for DVD media formats is implemented in a PC based system, additional mechanisms are used to enable the host to verify the integrity of the Media Key Block (MKB) and Media Identifier (ID_{media}) values it receives from the drive. To accomplish this, the following steps occur whenever either of these values is read:

- The host and drive carry out a mutual authentication and establish a secret shared key.
- The drive transfers the MKB or ID_{media} to the host, along with a corresponding message authentication code (MAC), which it generates using the shared key established during the authentication.
- The host uses the same shared key established during the authentication to generate a MAC for the received MKB or ID_{media} value, and verifies that it matches the received MAC.

Note that neither the MKB nor the ID_{media} needs to be kept confidential during transfer from the drive to the host. The purpose of the authentication and MAC calculation is only to enable the host to detect any alteration of those values by malicious software or hardware during the transfer.

The following subsections describe the authentication algorithm and MAC calculation function that are used, followed by the overall protocol used when the host requests an MKB or ID_{media} from a DVD drive.

4.4.1.1 Drive-Host Authentication

When implemented in a PC based system, CPRM for DVD media formats uses a procedure for bus authentication between the drive and the host, and the calculation of a shared key between the two. This procedure uses the same mechanisms that are defined for that purpose in the DVD-Audio Content Scramble System (CSS2), and a description is available separately. The execution of this procedure by both the drive and host for CPRM is referred to in this document as DVD Authentication (DVD_Auth).

4.4.1.2 Message Authentication Code Calculation

For implementing CPRM for DVD media in a PC based system, a mechanism is defined for calculating a message authentication code (MAC), for use when transferring the MKB and Media Identifier from drive to host. The mechanism is represented in this document by the function

DVD_MAC(m)

where m is a 64-bit message, and DVD_MAC returns an 80-bit MAC.

The DVD_MAC function is calculated following a successful DVD_Auth procedure, and uses data values saved from that procedure, including the shared key, as additional inputs. The DVD_MAC function makes use of CSS2-defined algorithms, and a description is available separately.

4.4.1.3 Protocols for Validating Media Key Block and Media Identifier

Figure 4-4 shows the protocol flow whereby a host acquires the Media Key Block (MKB) from a DVD drive, and verifies the integrity of the received value.

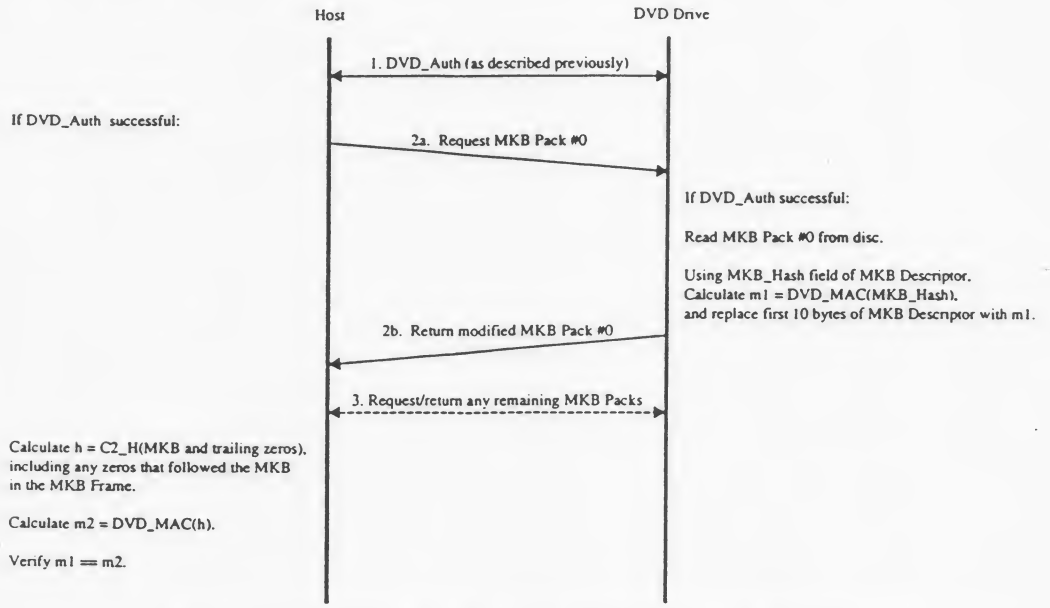

Figure 4-4 – Protocol Flow for Host Acquisition and Validation of MKB

The host shall verify the integrity of the received MKB (verify m1 == m2) before the calculation of the Media Unique Key (K_{mu}) indicated in step 2 of Sections 4.3.1.2.1 and 4.3.1.2.2. If the verification fails, the host shall abort the playback or recording session in progress. Note that whether the host verifies the MKB's integrity before or after the calculation of the Media Key (K_m) indicated in step 1 of those Sections is implementation defined.

Figure 4-5 shows the protocol flow whereby a host acquires the Media Identifier (ID_{media}) from a DVD drive, and verifies the integrity of the received value.

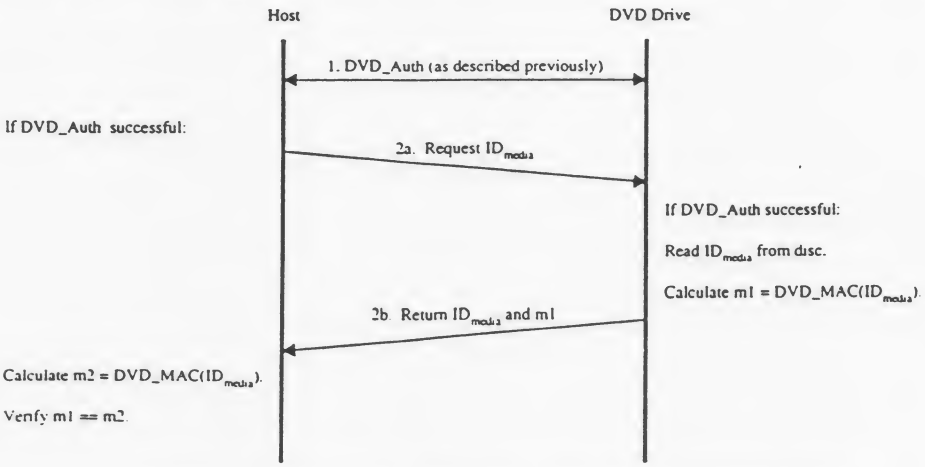

Figure 4-5 – Protocol Flow for Host Acquisition and Validation of ID_{media}

The host shall verify the integrity of the received ID_{media} (verify m1 == m2) before the calculation of the Media Unique Key (K_{mu}) indicated in step 2 of Sections 4.3.1.2.1 and 4.3.1.2.2. If the verification fails, the host shall abort the playback or recording session in progress.

4.4.2 Mt. Fuji DVD Command Extensions for CPRM

The Mt. Fuji specification defines commands and related structures used to control DVD devices (logical units). This section describes extensions incorporated into the Mt. Fuji 4 specification for logical units that support CPRM functionality. Some additional information that is not found in the Mt. Fuji 4 specification is also given, including the precise format of CPRM data values returned by the logical unit.

Note that the Mt. Fuji 4 specification uses different names for CPRM common components than those used elsewhere in this document. Specifically, the Mt. Fuji 4 specification refers to the Media Key Block as the Disc Key Block, and the Media Identifier as the Protected Disc Identifier. This is done in order to follow Mt. Fuji naming conventions, as well as to prevent conflicts with other component names already established in the Mt. Fuji document.

4.4.2.1 DVD CPRM Feature

The Mt. Fuji 4 specification defines a number of Features, which are sets of commands, mode pages, and behaviors or operations supported by a logical unit. Features implemented by a logical unit are reported to the host via the GET CONFIGURATION command. This command can be used to identify all possible Features, as well those Features that are current (i.e. currently available, which may depend on factors such as the type of media currently loaded).

A DVD Feature for CPRM is defined as shown in Table 4-11.

Table 4-11 – DVD CPRM Feature

Feature Code	Feature Name	Description	Mandatory Commands
$010B_{16}$	DVD CPRM	Ability to perform CPRM key management	REPORT KEY, SEND KEY, READ DVD STRUCTURE (Format Codes 06_{16} and 07_{16})

The DVD CPRM Feature Descriptor, obtained via the GET CONFIGURATION command, is shown in Table 4-12.

Table 4-12 – DVD CPRM Feature Descriptor

Bit / Byte	7	6	5	4	3	2	1	0
0	(msb)							
1	Feature Code = $010B_{16}$							(lsb)
2	Reserved			Version			Persistent	Current
3	Additional Length = 04_{16}							
4	Reserved							
5	Reserved							
6	Reserved							
7	CPRM version							

The **Current** bit, when set to zero, indicates that this Feature is not currently active. When set to one, the Feature is active. The DVD CPRM feature shall be active if and only if a CPRM compliant DVD disc is loaded. For DVD-RAM discs, this is defined to be true if the CPR_MAI table in the Control Data Area has a CPS_TY field value of 02_{16}. CPRM compliance for other DVD physical formats is not yet been defined.

The Feature Descriptor's **CPRM version** field shall be set to the value of the CPRM Version field of the CPR_MAI table.

The other fields of the Feature Descriptor shall be set as described in the Mt. Fuji 4 specification.

4.4.2.2 REPORT KEY Command Extensions

The REPORT KEY Command requests the start of the authentication process, and provides data necessary for authentication and for generating a Bus Key for the DVD Logical Unit.

Table 4-13 – REPORT KEY Command

Bit / Byte	7	6	5	4	3	2	1	0
0	Operation code ($A4_{16}$)							
1	LUN (Obsolete)			Reserved				
2	(msb)							
3								
4				Reserved/Logical Address				
5								(lsb)
6	Reserved							
7	Key Class							
8	(msb)							
9				Allocation Length				(lsb)
10	AGID			Key Format				
11	Vendor-Specific			Reserved		NACA	Flag	Link

The **Key Format** field indicates the type of information that is requested to be sent to the host.

Since a Bus Key is used in the transfer of the Media ID and Media Key Block, a new Key Format is defined for requesting an Authentication Grant ID for use in authentication prior to the transfer of those values, as shown in Table 4-14.

Table 4-14 – Key Format Code Definition for Requesting an AGID for CPRM

Key Format	Returned Data	Description	AGID Use
010001_2	AGID for CPRM	Returns an AUTHENTICATION GRANT ID for Authentication for CPRM (DVD_Auth)	Reserved & N/A

Table 4-15 shows the format of the data returned by the REPORT KEY command when Key Format 010001_2 is used.

Table 4-15 – REPORT KEY Data Format (with Key Format = 010001_2, Key Class = 0)

Byte \ Bit	7	6	5	4	3	2	1	0
0	(msb)							
1	REPORT KEY Data Length (06_{16})							(lsb)
2	Reserved							
3	Reserved							
4	Reserved							
5	Reserved							
6	Reserved							
7	AGID		Reserved					

This Key Format requests the logical unit to return an Authentication Grand ID for CPRM. After a CPRM Authentication Grant ID is obtained, the DVD_Auth procedure referred to in Section 4.4.1.1 can be carried out, by using the REPORT KEY and SEND KEY commands in the same way as described in the Mt. Fuji and CSS2 specifications. The resulting Bus Key is used in the validated transfer of *only* the Media Identifier and Media Key Block (and *not* any other data such as CSS2-related key data), as described in Section 4.4.1.3 and as supported by the READ DVD STRUCTURE command extensions described in the next section.

4.4.2.3 READ DVD STRUCTURE Command Extensions

Logical units that implement the DVD CPRM Feature support extensions to the READ DVD STRUCTURE command. The READ DVD STRUCTURE command, shown in Table 4-16, requests that the Logical Unit transfer data from the DVD Lead-in Area to the host.

Table 4-16 – READ DVD STRUCTURE Command

Byte \ Bit	7	6	5	4	3	2	1	0
0	Operation code (AD_{16})							
1	LUN (Obsolete)			Reserved				
2	(msb)							
3	Address							
4	Address							
5	Address							(lsb)
6	Layer Number							
7	Format							
8	(msb)							
9	Allocation Length							(lsb)
10	AGID		Reserved					
11	Vendor-Specific		Reserved			NACA	Flag	Link

The **Format** field indicates the type of information that is requested by the host. New **Format** values defined for CPRM are shown in Table 4-17, along with corresponding usage of the **Layer Number** and **Address** fields. Note that the Mt. Fuji 4 specification uses different names for CPRM components than those used in this document. This is done in order to follow Mt. Fuji naming conventions, as well as to prevent conflicts with other component names already established in the Mt. Fuji document.

Table 4-17 – CPRM Format Code definitions for READ DVD STRUCTURE command

Format Code	Returned Data	Layer Number Field Usage	Address Field Usage	Description
06_{16}	Protected Disc Identifier	Reserved	Reserved	Returns the CPRM Media Identifier and a validating MAC.
07_{16}	Disc Key Block	Reserved	Pack number	Returns the CPRM Media Key Block Pack data and a validating MAC.

For **Format** code 06_{16}, or format code 07_{16} with the **Address** field set to 00000000_{16}, the returned data includes a MAC that is calculated using a Bus Key, as described in Section 4.4.1.2. The host establishes the Bus Key via the DVD_Auth procedure, using the same command sequence that is used for CSS2 (via the REPORT KEY and SEND KEY commands, beginning with REPORT KEY Key Format = 010001_2) prior to calling the READ DVD STRUCTURE command. The READ DVD STRUCTURE command **AGID** field identifies the Authentication Grant ID that was used in establishing the Bus Key.

For **Format** code 07_{16}, the **Address** field is used to specify which MKB Pack is to be read. This field enables the host to read a Media Key Block Frame contained in multiple Packs.

The other fields of the READ DVD STRUCTURE command shall be set as described in the Mt. Fuji 4 specification.

4.4.2.3.1 PROTECTED DISC IDENTIFIER (Format 06_{16})

Table 4-18 – READ DVD STRUCTURE Data Format (With Format Field = 06_{16})

Bit / Byte	7	6	5	4	3	2	1	0
0	(msb)			DVD STRUCTURE Data Length (12_{16})				
1								(lsb)
2	(msb)			ID_{media}				
.								
9								(lsb)
10	(msb)			DVD_MAC(ID_{media})				
.								
19								(lsb)

This Command with this Format Code is used to carry out steps 2a and 2b of the protocol flow shown in Figure 4-5 of Section 4.4.1.3.

The **DVD STRUCTURE Data Length** field specifies the length in bytes of the following DVD STRUCTURE data that is available to be transferred to the Host. The **DVD STRUCTURE Data Length** value does not include the **DVD STRUCTURE Data Length** field itself. For a Format Code of 06_{16}, the value of this field is 12_{16}.

Bytes 2 through 9 return the ID_{media}. Bytes 10 through 19 return a MAC that is calculated using the function referred to in Section 4.4.1.2, as follows:

DVD_MAC(ID_{media}).

When the loaded disc is not CPRM compliant media, this command with Format = 06_{16} shall be terminated with CHECK CONDITION Status, 5/6F/01 COPY PROTECTION KEY EXCHANGE FAILURE – KEY NOT PRESENT.

When the DVD Logical Unit is not in the Bus Key state, this command with Format = 06_{16} shall be terminated with CHECK CONDITION Status, 5/6F/02 COPY PROTECTION KEY EXCHANGE FAILURE – KEY NOT ESTABLISHED.

4.4.2.3.2 DISC KEY BLOCK (Format 07_{16})

Table 4-19 – READ DVD STRUCTURE Data Format (With Format Field = 07_{16})

Bit / Byte	7	6	5	4	3	2	1	0
0	(msb)							
1	DVD STRUCTURE Data Length (7002_{16})							(lsb)
2	Reserved							
3	Total Packs							
4	(msb)							
.	DISC KEY BLOCK Pack Data							
28,675								(lsb)

This Command with this Format Code is used to carry out steps 2a and 2b of the protocol flow shown in Figure 4-4 of Section 4.4.1.3.

The **DVD STRUCTURE Data Length** field specifies the length in bytes of the following DVD STRUCTURE data that is available to be transferred to the Host. The **DVD STRUCTURE Data Length** value does not include the **DVD STRUCTURE Data Length** field itself.

The **Total Packs** field reports the total number of MKB Packs that are available for transfer to the host, which is the value of the Total MKB Packs Used field of the CPR_MAI table in the Control Data Area. The **Address** field in the command specifies which MKB Pack is read by the current command.

The **DISC KEY BLOCK Pack Data** field returns the Data field of the requested MKB Pack. For the first Pack only (command Address field = 00000000_{16}), the drive modifies the first 10 bytes of the Pack before returning it to the host. Specifically, the first 10 bytes of the MKB Descriptor are replaced with a MAC of the original MKB Descriptor's MKB_Hash field, as shown in Table 4-20.

Table 4-20 – Modified MKB Descriptor, as Returned by Drive to Host

Bit / Byte	7	6	5	4	3	2	1	0
0	DVD_MAC(MKB_Hash)							
...								
9								
10	00000_{16} (last part of Reserved field)							
11								

The first 10 bytes contain a MAC that is calculated using the function referred to in Section 4.4.1.2, as follows:

DVD_MAC(MKB_Hash).

When the loaded disc is not CPRM compliant media, this command with Format = 07_{16} shall be terminated with CHECK CONDITION Status, 5/6F/01 COPY PROTECTION KEY EXCHANGE FAILURE – KEY NOT PRESENT.

When the DVD Logical Unit is not in the Bus Key state, this command with Format = 07_{16} shall be terminated with CHECK CONDITION Status, 5/6F/02 COPY PROTECTION KEY EXCHANGE FAILURE – KEY NOT ESTABLISHED.

Macrovision Analysis of the Recent DVD "DeCSS" Hack
(November 15, 1999)

Background

As the world's leading provider of video copy protection solutions for fifteen years, Macrovision understands and appreciates the use of technical measures to protect copyrighted content. By its very nature, commercially attractive content is vulnerable to theft, be it by professional pirates, by unauthorized consumer copiers or by hackers. However, the Hollywood studios and Macrovision have always maintained that the primary aim of commercially implemented copy protection is to keep honest people honest.

The copy protection ecosystem for DVD video includes two fundamental elements: CSS (Content Scrambling System) and Macrovision ACP (Analog Copy Protection). The CSS system was designed to implement an encryption technology that would secure the digital video content and prevent unlicensed DVD hardware manufacturers from building devices that could play proprietary video content. The Macrovision technology was selected as the de facto standard to prevent unauthorized VCR copying of the unencrypted analog playback video. The CSS system is an encryption technology; the Macrovision system is a digital-to-analog copy protection technology.

The most recent "DeCSS" hack has involved breaking the CSS 40-bit encryption algorithm, as well as discovering the majority of the DVD manufacturer unique "keys." The DeCSS hack has nothing to do with Macrovision's digital-to-analog copy protection technology. Fortunately for content owners, the DeCSS hack has proven to be more difficult and complicated to use than the average consumer is willing to put up with.

Macrovision and most content owners do not anticipate any serious market effect from the most recent "DeCSS" hack because of the fact that most consumers can do little more than store one movie title at a time on their PC, unless they want to make degraded MPEG-1 video CDs from the hacked DVD video. Consumers are further thwarted from utilizing the DeCSS hack on a widespread basis because of the bandwidth limitations to be able to distribute the hacked video over the Internet, the absence of cost-effective DVD recordable devices and discs, the absence of a soft DVD player which will playback the DVD .VOB files, and the absence of truly consumer-friendly software programs to remove all forms of copy protection already deployed.

Fighting piracy is a three-pronged effort – utilizing the best copy protection technology available, developing and enacting legislative initiatives that protect the right's holder's copyrights, and implementing education and enforcement programs that support the laws. Macrovision's video copy protection and computer software copy protection business groups are continuously monitoring the Internet for hack sites. Macrovision has dedicated engineering teams that are working to continuously improve the security of our copy protection technologies in anticipation of future hacks. Macrovision has continued deploying its legal and technical resources towards enforcing security and technical standards for our licensees, enforcing our contracts and technical standards with existing licensees, and enforcing our intellectual property against those who illegally employ our technology or who circumvent it. Macrovision has recently served "cease and desist" orders on US and European ISP's that host hacker sites.

Macrovision Comments on Recent CSS Hack

Why the CSS Hack is a minimal threat to copyright owners

With regard to the recently advertised CSS Hack, what do the CSS hackers get after they decrypt the DVD content? They will get clear (unencrypted) digital video content in several files that have .VOB extension. The total of these .VOB files can range in size from 4.7 GB to over 9 GB. Today's recordable DVD discs have, at best, 2.5GB capacity (or 5.2 GB for double-sided discs); therefore direct DVD copying is unfeasible. Furthermore, since recordable DVD discs cost in the vicinity of $25-$30 per disc, there is little economic incentive for a consumer to make a DVD copy. The hackers can only store a 4.7GB DVD file in their computer's hard disk. Even with the 4.7GB recordable DVD drives that are expected to hit the market next year, the hackers will only get the linear version of the DVD movies which means no navigation capability (such as menu, chapters, interaction between chapters). The hackers would need to develop a DVD player that can play the linear DVD content. **Note that in all cases, the decrypted linear DVD content will contain Macrovision copy protection trigger bits which are used to prevent unauthorized analog VCR copies.**

Currently, there are no cost-effective methods of re-distributing the 'hacked' content – either in packaged media or Internet transmissions. Cost-effective recordable DVD hardware will not be widely available in the consumer market for another 2-3 years. Additionally, the new recordable DVD formats most likely will not be backward compatible with existing DVD players or PC-DVDs. Without recordable DVD discs, it will take multiple CD-ROMS (or other equivalent magnetic or optical storage media) to store or distribute a hacked DVD movie. If CD-ROMs are used it is likely that the consumer would have to convert the hacked DVD MPEG-2 streams to MPEG-1 video CD format, which is inferior in quality to the DVD MPEG-2 format. Transferring and/or downloading a hacked DVD movie from the Internet would take over 190 hours using a standard 56kb modem operating at maximum transfer rate, or over 10 hours using a 1 Mcgabit/sec ADSL line.

Why Macrovision digital-to-analog copy protection is important

The CSS hack in no way diminishes the need for effective Digital-to-Analog copy protection provided by Macrovision. There are now almost 600 million VCRs worldwide and the VCR sales continue to be robust throughout the world. Recent reports from the Consumer Electronics Manufacturers' Association (CEMA) show that for 1999, the rate of VCR sales in the U.S. is running 26% ahead of the record pace set in 1998, and there are expected to be in excess of 19 million VCRs sold in the U.S. in 1999. Almost 48% of all U.S. households have two or more VCRs. With blank videocassettes selling for well under $1.00, the VCR remains the easiest, cheapest, and most available device for consumers to make unauthorized copies for themselves, their friends, and their family.

The CSS hack does not affect the Macrovision DVD Copy Protection. As described above, CSS encryption is a totally different technology with a totally different purpose than Macrovision's DVD Digital-to-Analog Copy Protection technology. The Macrovision Copy Protection is not encrypted with CSS technology. Even though the CSS keys are hacked, the hacked movie content still contains the Macrovision Copy Protection (APS trigger bits). The APS trigger bits are set throughout the movie and would all need to be reset to entirely disable the analog copy protection.

Future watermarking/play control technologies will enhance copy protection security

Macrovision Comments on Recent CSS Hack

To counter future CSS hacking, 'watermarking' and 'play control' technologies are the true digital-to-digital copy protection solution – and one hopes that the CSS hack will expedite the pending CPTWG (Copy Protection Technical Working Group) industry-wide decision on watermarking. The Millennium Group (Macrovision, Philips, Digimarc) has proposed a robust watermarking/play control solution to the CPTWG that will address the shortcomings of the CSS encryption system.

In addition, the Digital Transmission Copy Protection (DTCP, i.e., secure 1394/USB) will help to ensure another form of copy protection by securing the transmission links between digital devices. In the future, watermarking will work in tandem with DTCP and with Macrovision's digital-to-analog copy protection to ensure a watertight ecosystem to protect rights owners' video content in all facets of digital and analog media, and digital and analog transmission.

Cautionary note on weakness of "remarking" watermarking approach

Macrovision believes that the CSS hack demonstrates a potential Achilles heel of the "remarking" technology that has been proposed by the Galaxy Group for 'copy-once' watermarking. The Millennium Group has proposed an encrypted "ticket" approach. A fundamental difference between the two competing solutions is that the remarking encoders will reside in millions of consumer devices, thereby increasing the exposure to hacking. The Millennium ticketing approach is inherently more secure since it relies on a single watermark that is encoded at the source and does not proliferate watermark encoders in all consumer devices.

Legislation and other future copyright protection technologies

By combining appropriate legislation with an industry-wide economic and secure watermark standard, the rights owners can further assure that copyright protection in the digital world will be more than adequate to "keep the honest consumer honest" and even to go a long way toward controlling professional piracy.

Other technologies that are anticipated to be available in the future and which will also assist in protecting content include CSS2, CPRM, and DVI-CP. Additional hardware and software security features that Intel and Microsoft will be embedding into microprocessors and operating systems in their future might also help to overcome CSS' shortcomings. In general, the PC security developments are being considered in the context of secure Internet communications. Macrovision intends to develop application software solutions that will work in this environment to further protect video and audio content – across as many varieties of digital media formats and distribution channels as are available.

MPA
MOTION PICTURE ASSOCIATION
Audiovisual Entertainment
For Global Audiences

EUROPEAN OFFICE
Avenue de Tervueren, 270-272
B-1150 Brussels, Belgium
Tel: (32-2) 778 27 11
Fax: (32-2) 778 27 50

If there is any problem with this transmission or if you receive this message in error, please contact the message sender stated below at (32-2) 778 27 11
Return facsimile number: (32-2) 778 27 50

To: Brad Hunt	
Company Name:	
Fax Number: (1-818) 461 1502	
From: Nalla Knight	
Secretary: Lucie-Lise	
Date: 3/11/00	**Re:**
Number of pages including cover sheet: 4	

Message:

As per your e-mail.

Kind regards,

Nalla

I.I.

M-19078